

Attorney Docket RSW920010100US1
Serial No. 10/043,439

Remarks

In the final Office Action mailed on December 1, 2005, claims 1, 2, 4, 6, 18-22, 26 and 38-43 were rejected under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent Application Publication No. 2002/0120717 A1 to Giotta. Claims 41-43 were rejected under 35 U.S.C. § 101 as being directed to non-statutory subject matter. Claims 3, 5, 7-17, 23, 25 and 27-37 were objected to as being dependent upon a rejected base claim but as otherwise reciting allowable subject matter. Remaining claim 24 was not specifically addressed in the final Office Action.

Claims 41-43 stand rejected under 35 U.S.C. § 101 as being directed to non-statutory subject matter. Per the suggestion set out in the Office Action, applicants has amended claim 43 by changing "computer usable medium" to --computer usable storage medium--. Claims 41 and 43 have been canceled. Accordingly, it is believed that these amendments overcome the § 101 rejection.

With this paper, claims 1, 21, 41 and 43 have been canceled. Claim 43 has been amended to overcome the § 101 rejection. Further, claims 2, 4, 22 and 24 have been rewritten in independent form. No new issues are raised by these amendments. Accordingly, entry of this paper is respectively requested.

The Giotta application discloses a message system comprising Message Managers MM_a, MM_b and Client Managers CM, see ¶ 39. The CM's are responsible for managing client connections, forwarding messages from produccr clients to an MM and forwarding messages from the MM to a consuming client, see ¶ 21. The MM's are responsible for storing and delivering messages, see ¶ 39 and the abstract. The messages are stored in destinations existing on one or more MM's. When a destination exists on more than one MM, one MM is designated as the primary, while all other MM's containing the destination are backups. The backups do not provide any services unless the primary fails, see ¶ 22. A multicast protocol is provided such that data is distributed to the primary and backup MM's without incurring significantly more network traffic than there would be if no backup MM's existed, see ¶ 25.

Attorney Docket RSW920010100US1
Serial No. 10/043,439

The Giotta application teaches in ¶ 140, “[c]ommitting a transaction is the crucial last step when all of the individual actions that comprise the transaction are effectively carried out.”

The Giotta application further teaches in ¶ 140 through ¶ 155:

In a 2 phase commit, one entity acts as the transaction manager. In the first phase the transaction manager requests a guarantee that each of the transaction participants is capable of successfully executing the transaction. In the second phase, the transaction manager instructs the participants to actually perform the commit or, if not all participants were able to offer a guarantee of success, to rollback the transaction. JMS transactions must occur within one session, and they encompass all messaging activity that has occurred within that session since the last commit. For this reason the session tasks in the CM act as transactions managers. The transaction participants are all of the MM destinations with which that session has interacted during the current transaction. Transaction management is a common piece of functionality that may be employed by all session tasks. For this reason it is depicted as a separate box in the Services module in drawing 3 which shows the internal structure of the CM.

[0141] The specific steps executed in the processing of a transaction are:

[0142] Produce Message: This occurs in a fashion similar to the non-transacted case. The producer sends the message and continues processing without waiting for a reply from the server. The CM passes the message to the appropriate MM, where it is stored marked as uncommitted. The CM adds the message ID to the list of produced messages for the open transaction of the corresponding session.

[0143] Consume Message: The MM sends a message to the CM, which forwards it to a consumer. The CM adds the message ID to the list of consumed messages for the open transaction of the corresponding session. The message continues to be stored in the MM where it is locked until the MM received either a commit (equivalent to an ACK) or a rollback.

[0144] Commit: The list of produced and consumed message IDs for a session should be organized by destination. The CM sends a COMMIT command containing the lists of produced and consumed message ID's for all destinations. The list of consumed message ID's is that which is provided by the client. The one stored in the session may contain messages that have not yet been delivered to the consumer. If only one

Attorney Docket RSW920010100US1
Serial No. 10/043,439

destination is involved, this may be a 1 phase commit, and the CM may synchronously wait until the reply from that destination arrives. If more than one destination is involved then a 2 phase commit is needed. See below for more details.

[0145] Rollback: The CM sends a ROLLBACK command containing the lists of produced and consumed message IDs for that destination. The list of consumed message ID stored in the session is used, as the message store should be returned to the state it had at the beginning of the transaction.

[0146] Two Phase Commit:

[0147] A simple two phase commit protocol may be used to commit transactions across multiple destinations. The requirements of JMS transactions are less demanding than those of many other transactional systems. Transactions occurring in different session have no interdependencies and since one producer may not produce in more than one session, JMS sets no restrictions on the relative ordering of messages from different transactions.

[0148] The CM, which handles the session that is conducting the transaction, acts as the transaction manager. The steps of a 2-phase commit are:

[0149] COMMIT_PREPARE command request is sent to all MMs and lists all of the destinations involved in the transaction and the id's of the consumed and produced messages per destination, as well as a unique transaction ID.

[0150] The Destination Command Distributor distributes copies of the command to each destination that is involved in the transaction.

[0151] Each destination checks that all produced messages for which it is responsible are available in the message store and have uncommitted state. It checks that all consumed messages for which it is responsible are in the message store and are locked by the session of the transaction. If so, it sends a reply containing COMMIT_READY and a list of destinations. Otherwise it sends a COMMIT_FAIL message. If the MM has no destinations involved in the transaction, then it sends a COMMIT_READY message containing no destinations.

[0152] If the CM receives COMMIT_READY from all involved MM's, then it sends a COMMIT_FINAL message to the transaction channel,

Attorney Docket RSW920010100US1
Serial No. 10/043,439

containing the transaction ID.

[0153] The Commit Manager in each MM forwards the COMMIT_FINAL message to each destination involved. Each destination changes the state of the committed messages and returns COMMIT_COMPLETE. If the MM has no destinations involved in the transaction, then it sends a COMMIT_COMPLETE directly.

[0154] After all COMMIT_COMPLETE messages have been received, the CM returns a success message to the client.

[0155] If the CM receives one or more COMMIT_FAIL messages in response to the COMMIT_PREPARE, or one or more of the destinations times out, then it sends COMMIT_ROLLBACK messages to all involved destinations and notifies the client of failure.

With regard to claims 2, 4, 22 and 24, the Office Action indicates that “sending a COMMIT_PREPARE task” locks “uncommitted messages in the storage.” It is believed that the generation of a COMMIT_PREPARE command does not result in consumed messages being locked. Instead, after an MM sends a message to a CM, the MM continues to store this message until the MM receives either a commit or a rollback, see ¶ 143. Hence, storage of the message by the MM is not initiated in response to a COMMIT_PREPARE command, but, instead, is believed to occur in response to the sending of the message by the MM to a CM. Accordingly, it is submitted that claims 2-5 and 22-25 define patentable invention over the Giotta application.

Claim 6 recites:

A computer implemented method for processing shared data comprising:
receiving a work item message from a messaging service by
consuming the work item message from a topic of the messaging service,
wherein the topic is a category by which messages in the messaging
service are sorted;
processing the work item message based on the topic; and
publishing a result to a result topic of the messaging service,
wherein the result topic is a category identifying results of processing the
work item message.

Claims 26 and 42 recite similar subject matter in system and computer program product formats.

Attorney Docket RSW920010100US1
Serial No. 10/043,439

With regard to claims 6, 26 and 42, the Office Action indicates that Giotta teaches providing a work item message comprising a COMMIT_PREPARE command from a topic. The Office Action further states that "in JMS the topics identify a destination" and "each destination [is] (aka [a] topic in the JMS system)." In Giotta, the COMMIT_PREPARE command is sent by a CM to all MMs, see ¶ 149. Hence, it is believed that the COMMIT_PREPARE command is not consumed from a topic or destination, see claims 6, 26 and 42, but, rather, is sent to all MM's in which the destinations are found, see paragraphs 22, 148 and 149. Further, a CM sends the COMMIT_PREPARE command to all MM's, see ¶ 149; hence, it is believed that a CM does not perform a sorting function. Accordingly, it is submitted that Giotta does not disclose, teach or suggest the subject matter set out in independent claims 6, 26 and 42, and dependent claims 7-17 and 27-37.

Claim 18 recites:

A computer implemented method for processing shared data comprising:
receiving a result from a result topic of a messaging service,
wherein the result topic is a category identifying results of processing
work item messages;
processing the result based on a type of the result; and
updating shared data based on a brand of the result, wherein the
brand of the result identifies one of a current node and all nodes.

Claims 38 and 43 recite similar subject matter in system and computer program product formats.

With regard to claims 18 and 38, the Office Action indicates that COMMIT_COMPLETE and COMMIT_ROLLBACK messages, see paragraphs 153-155 in the Giotta application, are equivalent to a result from a result topic. The Office Action also indicates a COMMIT_COMPLETE result is processed by "marking messages as complete and updating the data state." The Office Action further indicates that data is updated based on a brand, i.e., the ID's of the consumed and produced messages per destination, of the result. If "processing the result based on a type of the result," as recited in claim 18 (similar limitations are set out in claim 38) involves "marking messages as complete and updating the data state," as alleged in the

Attorney Docket RSW920010100US1
Serial No. 10/043,439

Office Action, then it appears that no additional step of "updating shared data based on a brand of the result," as recited in claim 18 (similar limitations are set out in claim 38), occurs as the data has already been updated, i.e., the state of the committed messages has been changed, see ¶ 153. Accordingly, it is submitted that claims 18 and 38 and dependent claims 19, 20, 39 and 40 define patentable invention over Giotta.

In view of the above remarks, applicants submit that claims 2-20, 22-40 and 42 define patentably over the prior art. Early notification of allowable subject matter is respectfully requested.

Respectfully submitted,
STEVENS & SHOWALTER, L.L.P.

By



Robert L. Showalter

Registration No. 33,579

7019 Corporate Way
Dayton, OH 45459-4238
Telephone: 937-438-6848
Fax: 937-438-2124
Email: rshowalter@sspatlaw.com